DESIGN AND FABRICATION OF THE AUTONOMOUS FRUIT PLUCKING ROBOT

Advait Patole [1] | Jeet Acharya [1]

[1] Sardar Patel College of Engineering, Mumbai, India.

ABSTRACT

The people of India are mainly dependent on the agriculture and allied field. In order to alleviate the crisis on farming fronts, it is necessary that we should discover new mechanisms by which farmers are adequately and rightfully compensated for their labour. That is the reason why an autonomous robot had to be made in order to reduce the labour and labour costs of farming in India. Fruit plucking is an important process in farming. An autonomous robot which could pluck fruits on its own in a farm would be the best possible alternative of manual labour for plucking fruits in a farm. Such a fruit plucking robot can move autonomously in a farm containing plantations of a given fruit using placards or signs placed at each plant bearing that particular fruit. After detecting the actual fruit using image processing, it can operate its links of the robotic arm to reach the particular fruit. After reaching the fruit, it can pluck the fruit using the plucking mechanism. This whole process can be done by a single robot autonomously which can be released in the farm to perform the fruit plucking process. It provides an excellent substitute of manual labour for performing the plucking process in agricultural fields.

KEYWORDS: robotic arm, hough circle transform, OpenCV DNN, inverse kinematics, pixel to co-ordinate conversion, serial communication.

INTRODUCTION:

There is a rapid rise in the population of our country. It becomes very difficult to feed such a huge population which is always increasing when the number of farmers and cultivating fertile land in the country is decreasing. Therefore, it is highly important to focus on various ways to substitute labour and at the same time increase productivity with reduced input revenue and energy. Also, while considering these aspects the environment must also be taken into consideration.

MATERIALS AND METHODS:

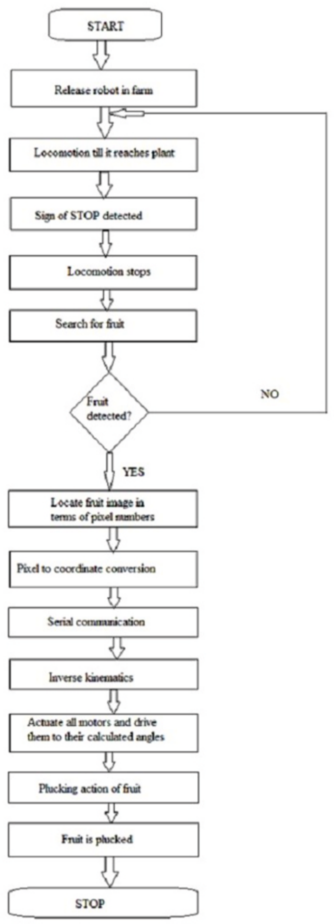

Figure 1: Flowchart depicting the actual plucking process of fruits in farms

Design of the robotic arm:

The design is basically similar to 3R Robotic Planer Arm. It contains 3 links which allow rotational movement about the joints. The joints are named as shoulder, elbow and the wrist with the claw at the end. The motors of required torques are attached at those joints. They are programmed so as to control the angle of the links so that the claw reaches the fruit by detecting the fruit coordinates in space via image processing. For plucking the fruit, the direct plucking mechanism was used.

The final design of the robotic arm on CATIA designing software is as shown here:-

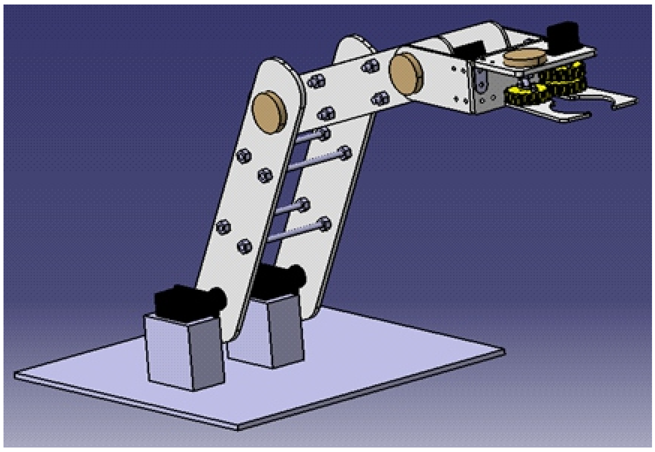

Figure 2: Design of the robotic arm of the fruit plucking robot

Salient features of the design:

- Its maximum range is approximately 42 cm. So it can pluck fruits within a circle of diameter of 42 cm.
- The robot is also versatile in the plane containing the z-axis (height) and the y-axis (perpendicular to plantations).
- The range of fruit sizes that the claw can hold is approximately 5cm to 7 cm and generally, majority of the fruits lie inside this range. Hence, this design was chosen as the final design for plucking the fruits.

Motor torque calculation and selection:

Using the values of the weight of the fruit to be plucked, the link lengths, the weight of the motors and miscellaneous things, the motor torques were calculated. Accordingly, the number of motors to be used at every joint and their selection of motor torques was done. Sufficient and appropriate factor of safety was also taken while selecting the motors.

These values of the calculated motor torques and the selected motor torques with the number of motors at a joint are specified in the table below:

Table 1: Table for calculated and selected motor torques at each joint

Sr. No.	Position of motor	Calculated torque (kg-cm)	No. of drive motors at that position	Selected Motor torque (kg-cm)
1.	Shoulder	24.3	2	15
2.	Elbow	12.76	1	15
3.	Wrist	4.5	1	12

Manufacturing of the robotic arm:

1. **Selection of material for links:** The robotic arm has to be made of a material that is strong enough to handle the stresses occurring during the static and dynamic condition of the arm. It has to be light weight so that it helps to reduce the torque required to lift the fruit. This would also help to reduce the cost spent on the motors. The material should be such that the machining of the complicated areas is simple and cost-effective. The material should be inexpensive as well. Acrylic is light weight, cheap, strong enough to handle the stresses and easily available in the market. Laser cutting provides accuracy and is comparatively cheaper than other accurate machining processes. Hence, acrylic of thickness 3mm was selected for the links of the robotic arm which was machined by laser cutting operation.

2. **Fabrication:** One side of each link is connected to the previous link through horns, which is fixed on one link and connected to the motor shaft of the other link. The other end is just supported on the dummy shaft that rotates freely about the shaft axis, which is fixed on one of the link. The motors of the shoulder are fixed on the wooden block that is connected to the base using aluminum supports.

   The manufactured robotic arm is shown below:

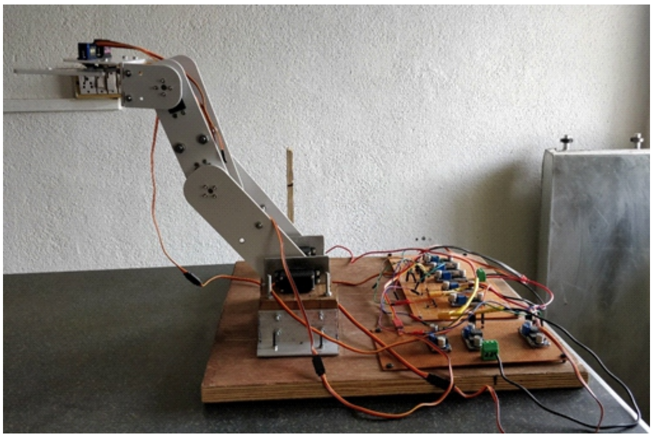

Figure 3: Left-hand side view of the manufactured robotic arm

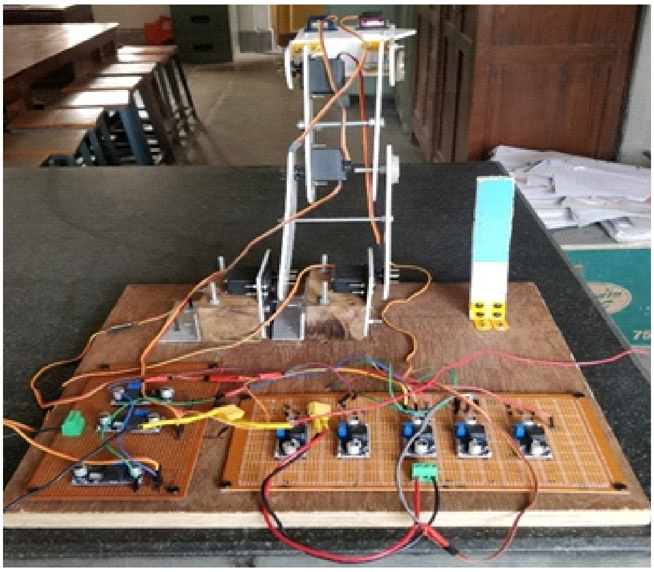

Figure 4: Back view of the manufactured robotic arm

Fruit detection:

It's difficult to run deep-learning libraries on raspberry pi, hence there is good news that OpenCV itself comes with its own deep-learning library that can run conveniently on raspberry pi's CPU power without any help of GPU's. OpenCV DNN[1] only requires OpenCV environment. OpenCV supports pre-trained models using Tensorflow and Caffe. It also supports various networks like YOLO, MobileNetSSD, Inception-SSD, Faster RCNN Inception [2], Faster RCNN[3] ResNet.

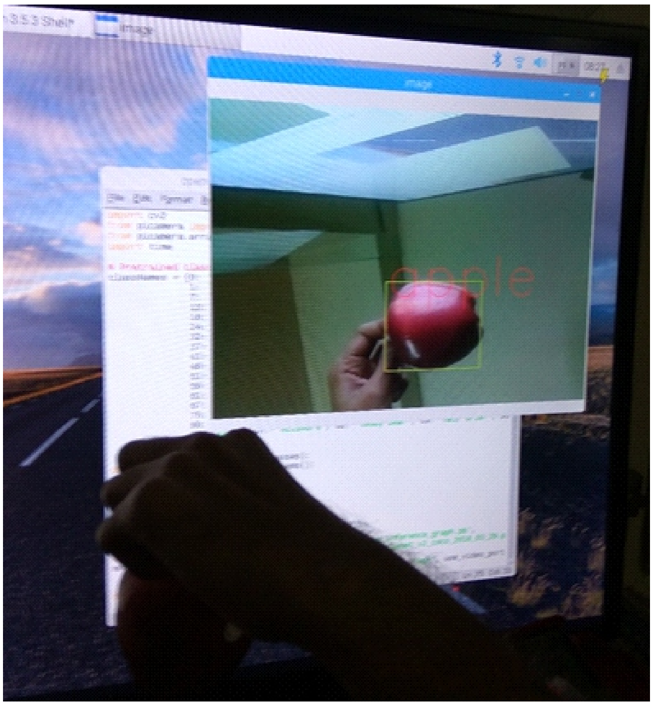

Figure 5: Apple detection using OpenCV DNN

Working model of the navigation robot:

The navigation robot is a regular 4-wheel robot-driven using motors.

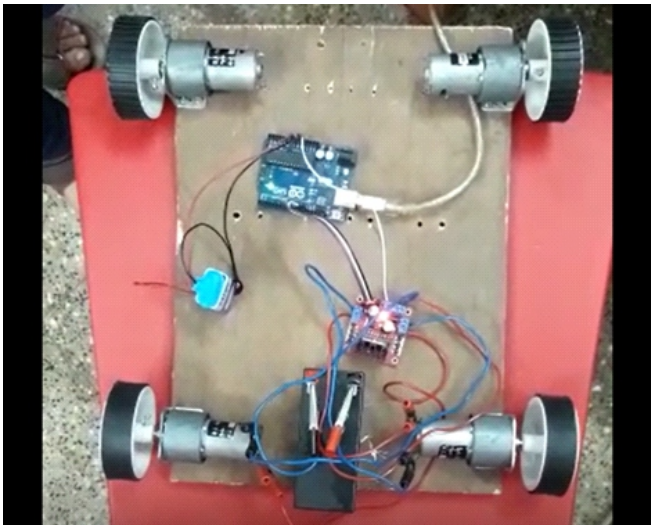

Figure 6: Top view of the working model of the navigation robot

Sign detection:

The navigation robot makes use of the traffic signs for navigation and the signs are used as directions for the robot traffic signs are taken as input which is processed using software and raspberry pi 3. The output of raspberry pi 3 is given to Arduino via serial communication [4] which further controls the motor driver (L298N) and the motor driver drives the motor. The raspberry pi recognizes various signs and understands what signs it is (FORWARD, RIGHT, LEFT, STOP)[5] using K-means clustering[6] and Hough circle transform.[7] The information that is collected by raspberry pi is passed on to Arduino using serial communication.

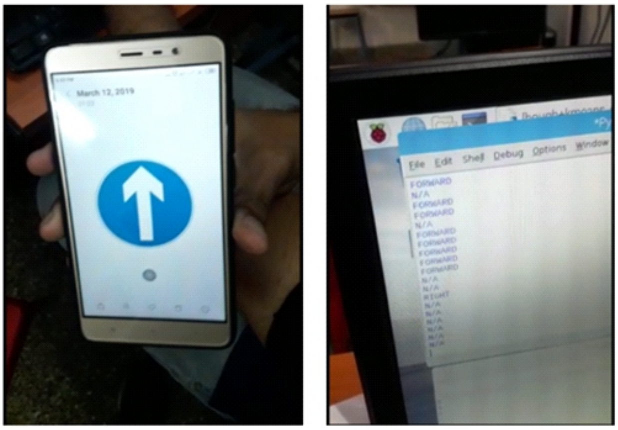

Figure 7: Sign detection

Serial communication:
Serial Communication is a process to send data in the form of bytes over a communication channel or computer bus. Serial communication is used to connect raspberry pi and Arduino to send information from raspberry pi to Arduino. The pixel co-ordinates of the fruit received after the detection is to be used for the robotic arm to pluck the fruit and data is transferred from raspberry pi to the Arduino. The Arduino is used for controlling the servo motors.

Pixel to coordinate conversion:
The object detection program detects the fruit which is in front of the camera and shows it on the computer screen. When the fruit is detected, the location of the top left corner of the box is within 0-640 pixels in X axis and within 0-480 pixels in Y axis. The exact location of the fruit can be obtained by converting these pixel values into actual space coordinates.

For calculation of the y coordinate there are two cases for a certain depth :

A] If the pixel value of the point in Y axes is greater than 240, then

Y-coordinate = ((pixel value in y direction)-240) * (44/240) * (depth/110) * (-1)

B] If the pixel value of the point in Y axes is less than 240, then

Y-coordinate = (240 - (pixel value in y direction)) * (44/240) * (depth/110) * (1)

Table for readings of test case depth against the actual depth of fruit for total pixel values is given below:

Table 2: Space co-ordinates after pixel conversion

Sr. No.	Cases	Depth (z) (cm)	Total pixel value in x-axis	Total pixel value in y-axis	Total actual length at given depth in x-axis (cm)	Total actual length at given depth in y-axis (cm)
1.	Test	110	640	480	118	88
2.	Actual	32	640	480	34	26
3.	Actual	20	640	480	21	16
4.	Actual	40	640	480	43	32

Inverse kinematics:
The obtained space coordinate values are used to calculate angle of the links using inverse kinematics. The claw reaches the fruit using the values of the angles of the links using inverse kinematics. Using inverse kinematics concept, the angles of the links can be calculated and the servo motors can be given those particular angles for actuation. Using those angle values, the claw reaches the fruit and does the plucking action.

RESULTS AND DISCUSSIONS:
After performing various trials on the fruit plucking process using the robot, it was observed that the robot could successfully reach the fruit and pluck it at its position.

The following figure shows the robot which is in action of plucking the fruit:

Figure 8: Plucking of the fruit by the robot

CONCLUSION:
The conclusions after successfully manufacturing the robot are:

1. The robot can move along the desired paths in the fields using different signs placed at appropriate locations.
2. The robot can pluck fruits on the plants in the farm successfully.
3. Small variations in the fruit colour, size or shape do not affect the performance of the robot.
4. After releasing this robot in the field, it can detect the fruits from the plants and pluck it along with its locomotion towards different plants of the plantation in the farms.
5. The paper proposes an automated system using the concepts of OpenCV for navigation of the robot and plucking the fruits.
6. The locomotion and the plucking action of the robot can be done simultaneously when it is released in a farm for plucking the fruits.

REFERENCES:

1. Roy, S. (2019). Real-Time Object Detection on Raspberry Pi Using OpenCV DNN. Retrieved from https://heartbeat.fritz.ai/real-time-object-detection-on-raspberry-pi-using-opencv-dnn-98827255fa60
2. Lv, J., Yang, F., Shen, G., & Ma, Z. (2019). Research on trunk and branch recognition method of apple harvesting robot - IEEE Conference Publication. Retrieved from https://ieeexplore.ieee.org/document/6273345
3. R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms. (2019). Retrieved from https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e
4. Raffin, A. Simple and Robust {Computer — Arduino} Serial Communication. Retrieved fromhttps://medium.com/@araffin/simple-and-robust-computer-arduino-serial-communication-f91b95596788
5. Traffic light and sign detection for autonomous land vehicle using Raspberry Pi - IEEE ConferencePublication. (2017). Retrieved from https://ieeexplore.ieee.org/document/8365328
6. Lau, B. Andrew Ng's Machine Learning Course in Python (Kmeans-Clustering, PCA). Retrieved from https://towardsdatascience.com/andrew-ngs-machine-learning-course-in-python-kmeans-clustering-pca-b7ba6fafa74
7. Hough Line Transform — OpenCV 3.0.0-dev documentation. Retrieved from https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_houghlines/py_houghlines.html